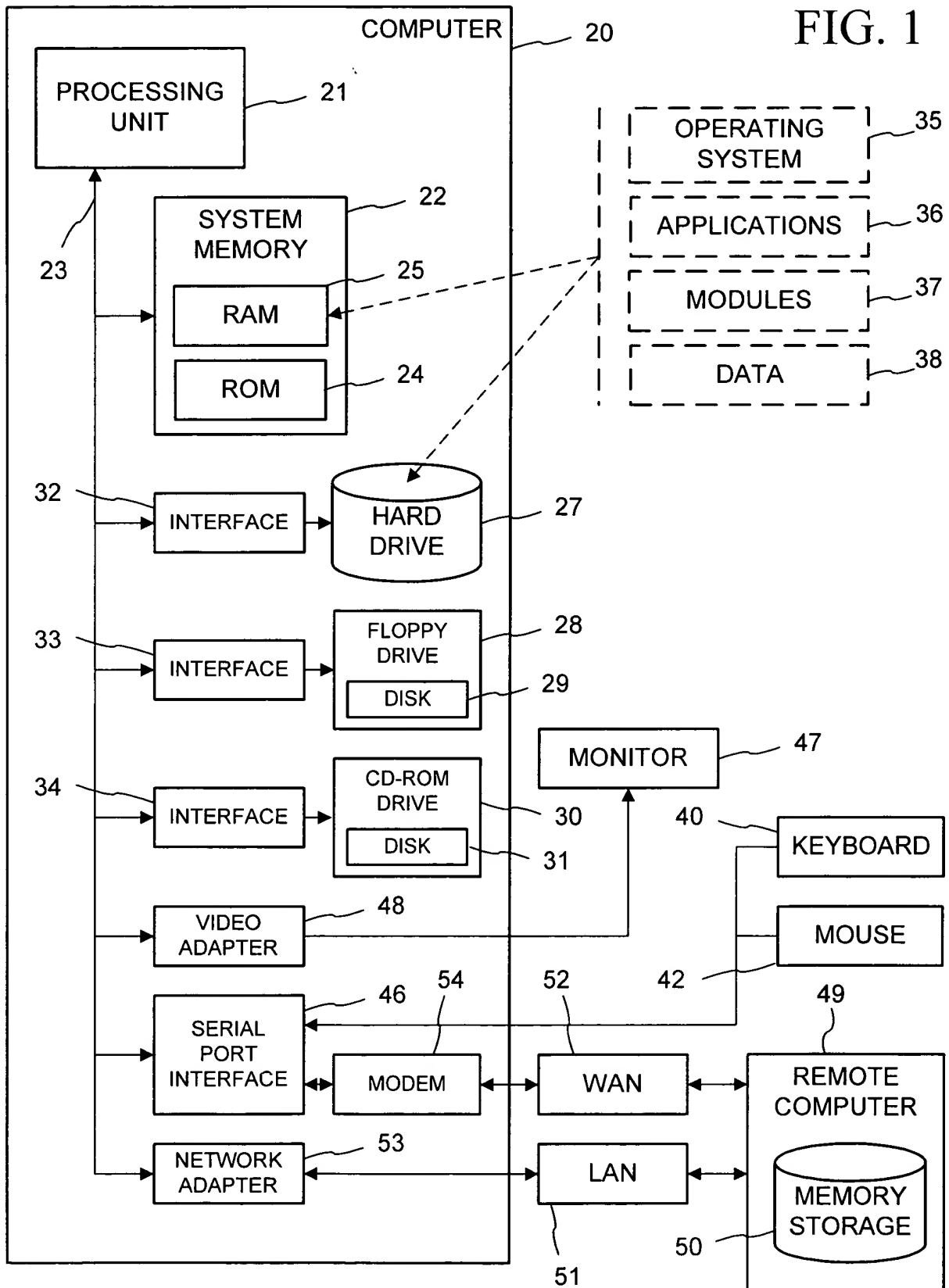


FIG. 1



0 1 2 3 4 5 6 7 8 9
 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29
 30 31 32 33 34 35 36 37 38 39
 40 41 42 43 44 45 46 47 48 49
 50 51 52 53 54 55 56 57 58 59
 60 61 62 63 64 65 66 67 68 69
 70 71 72 73 74 75 76 77 78 79
 80 81 82 83 84 85 86 87 88 89
 90 91 92 93 94 95 96 97 98 99

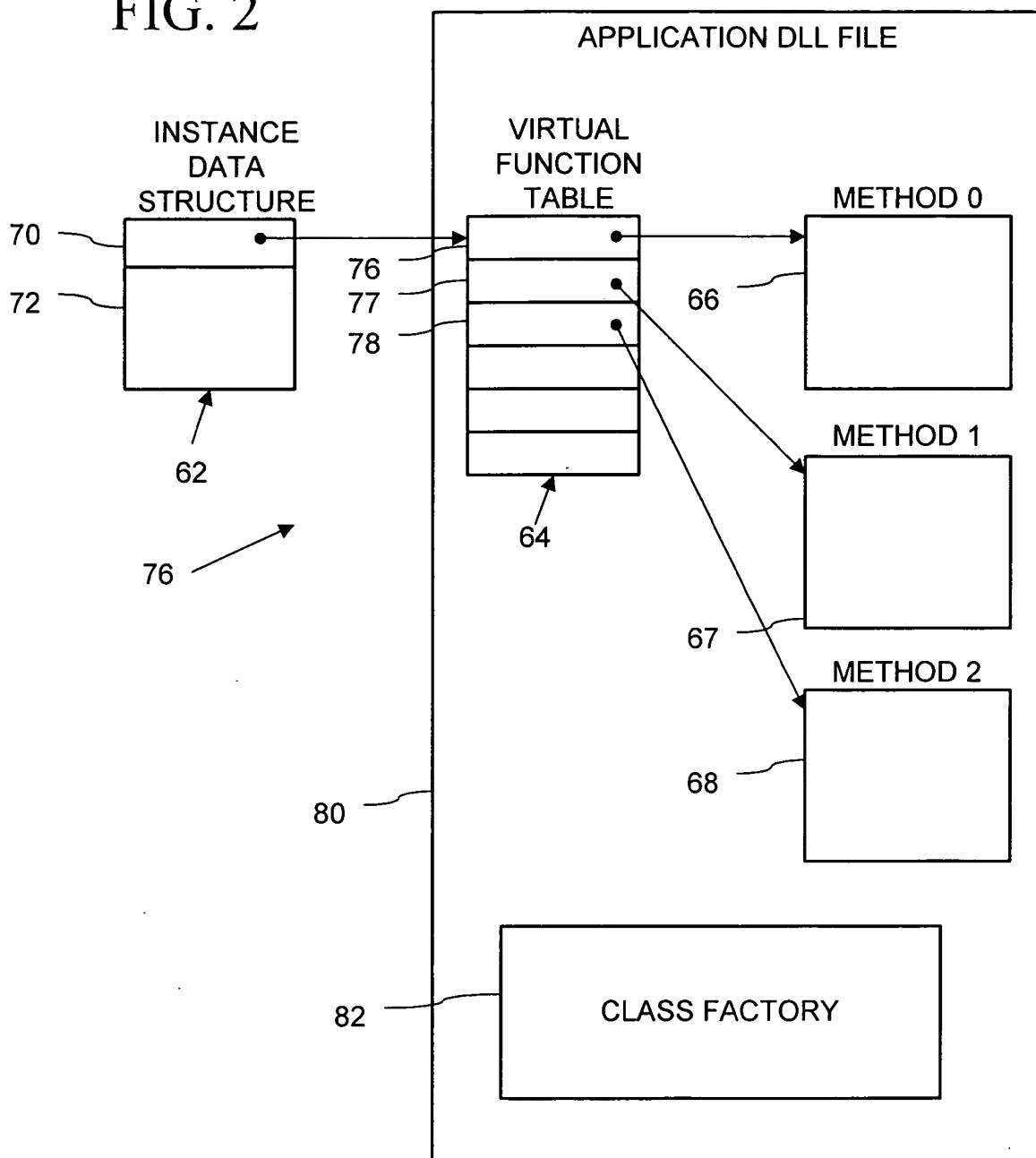


FIG. 3

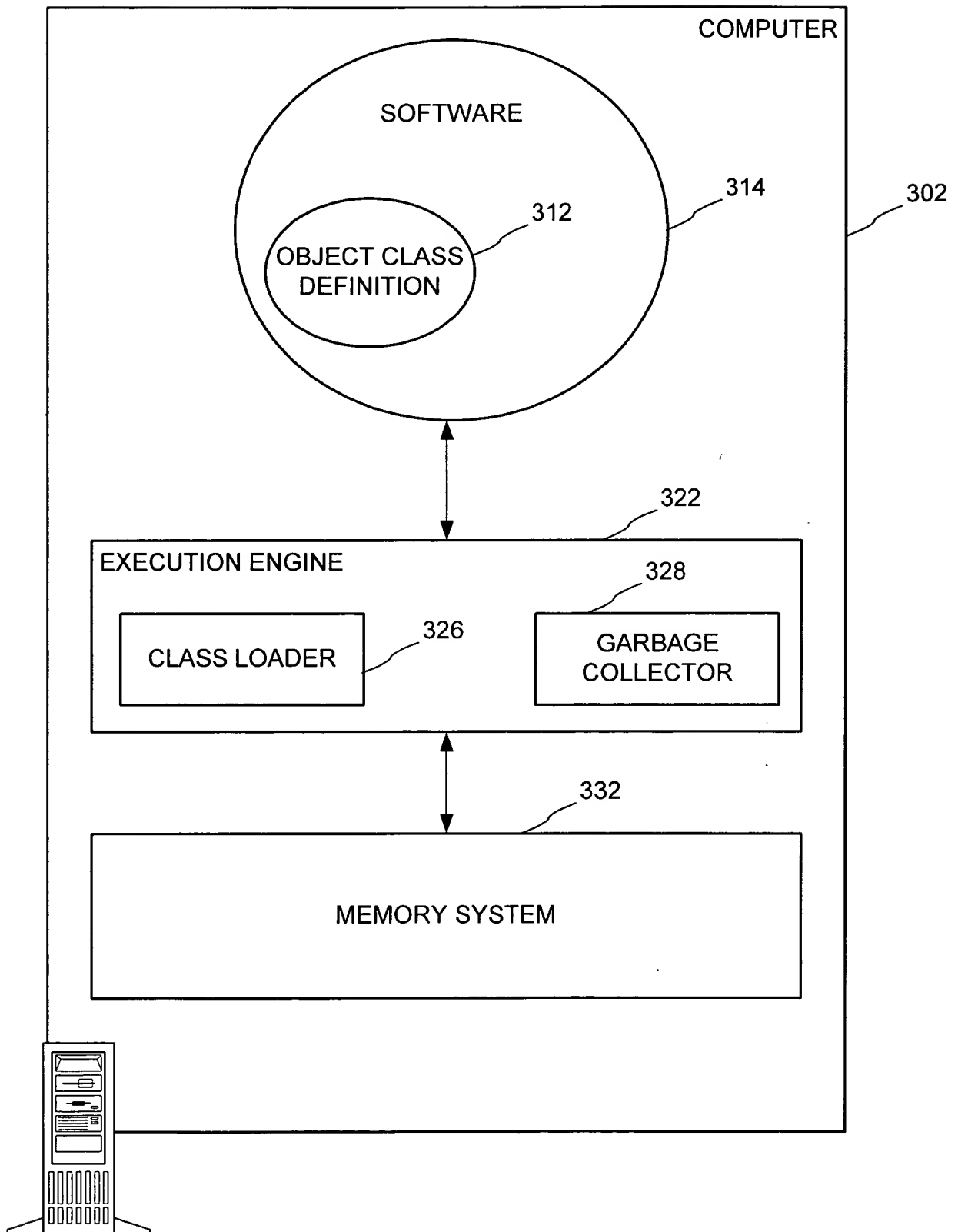


FIG. 4

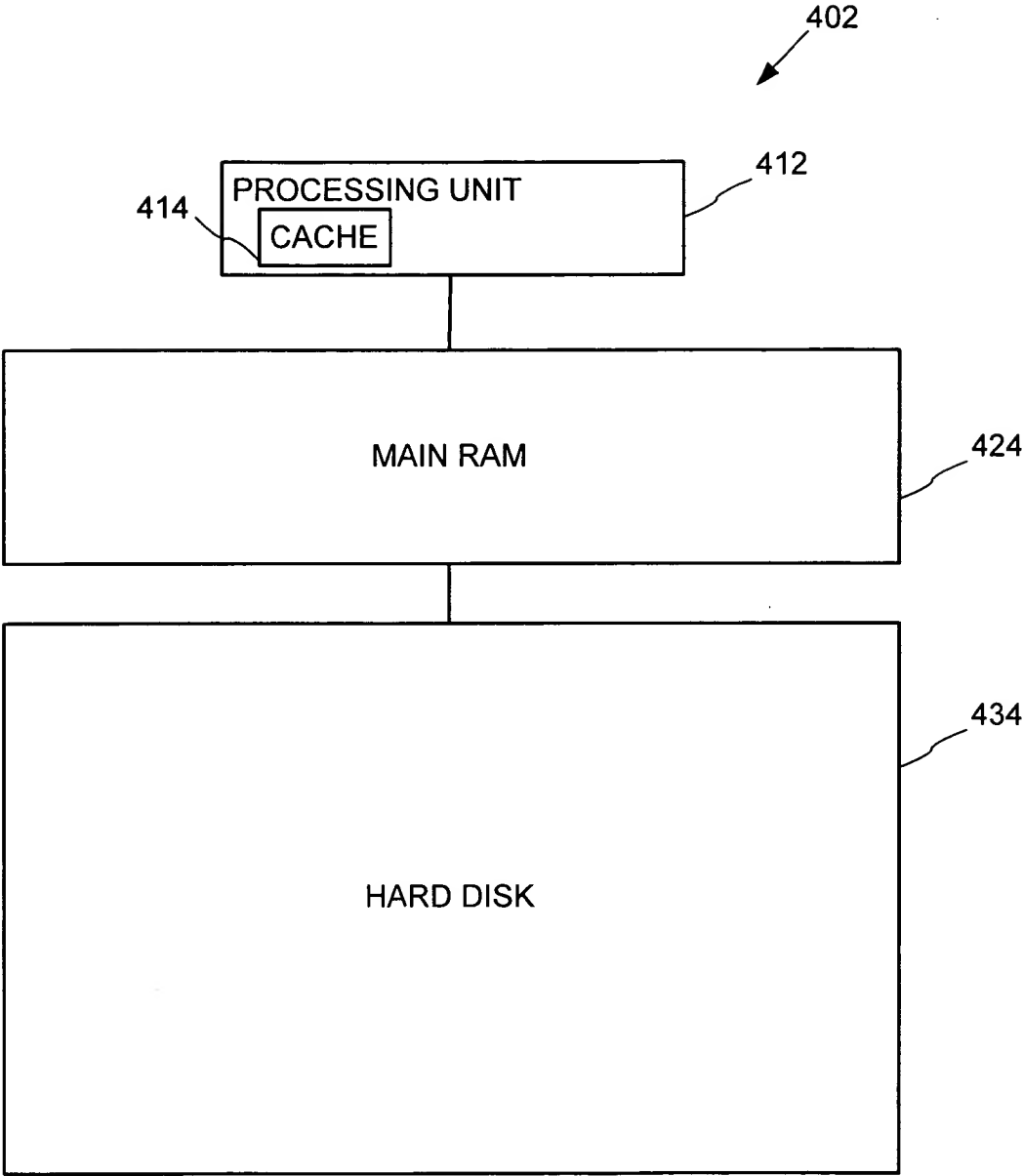


FIG. 5

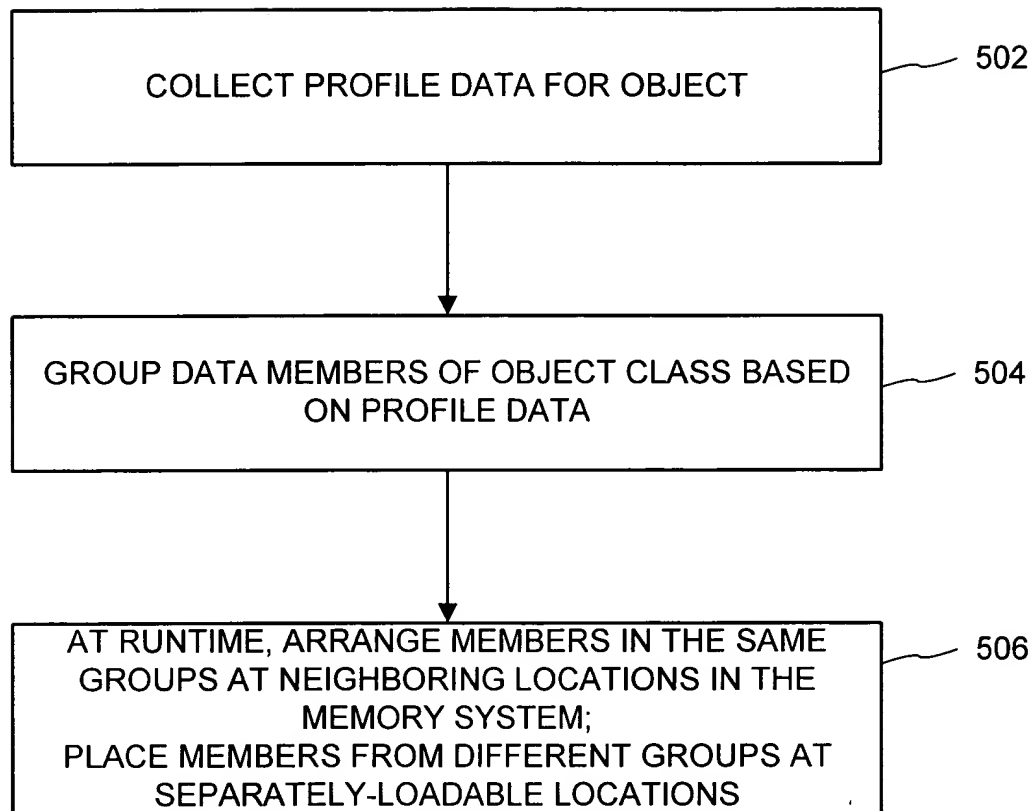


FIG. 6

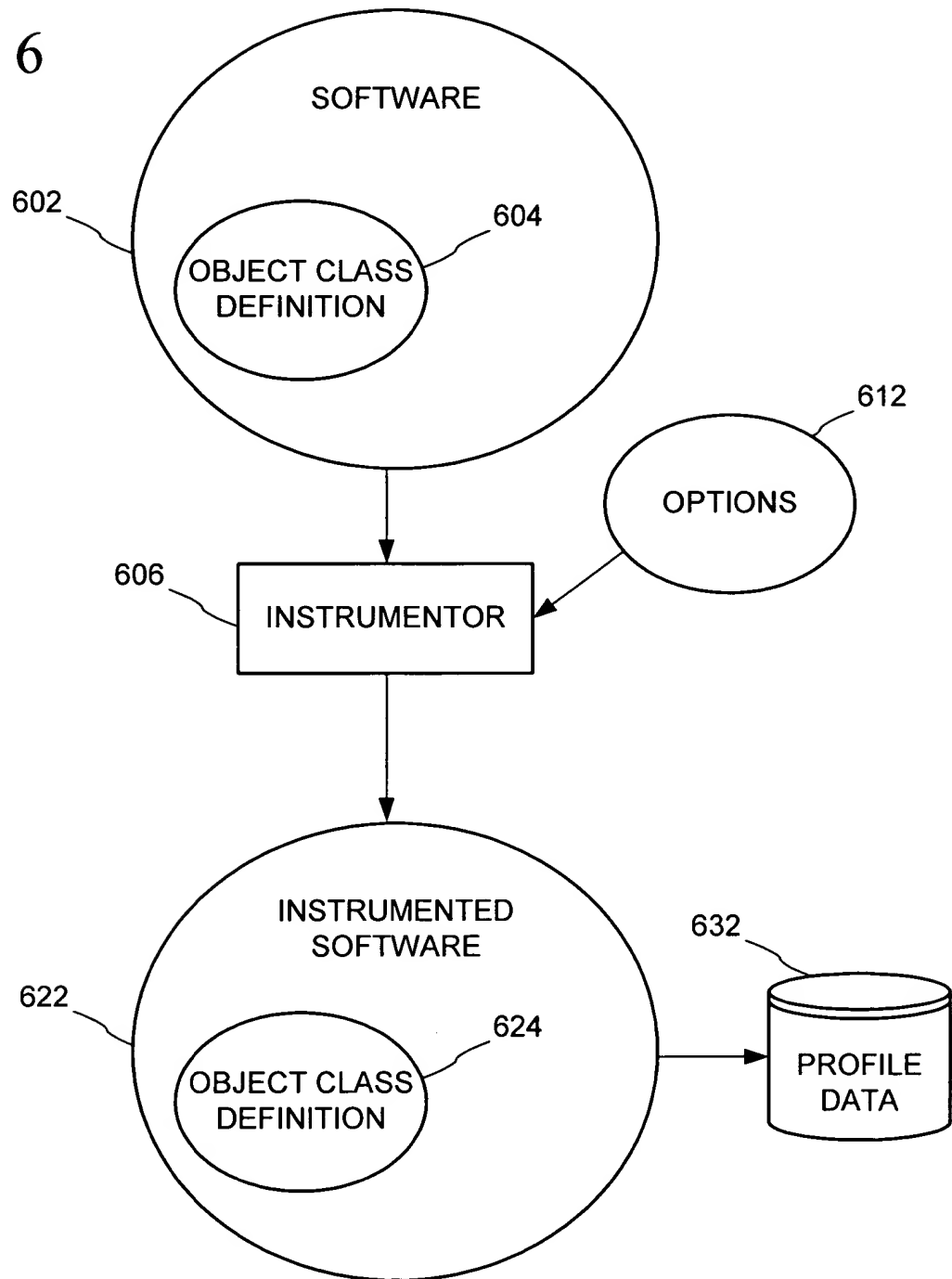


FIG. 7

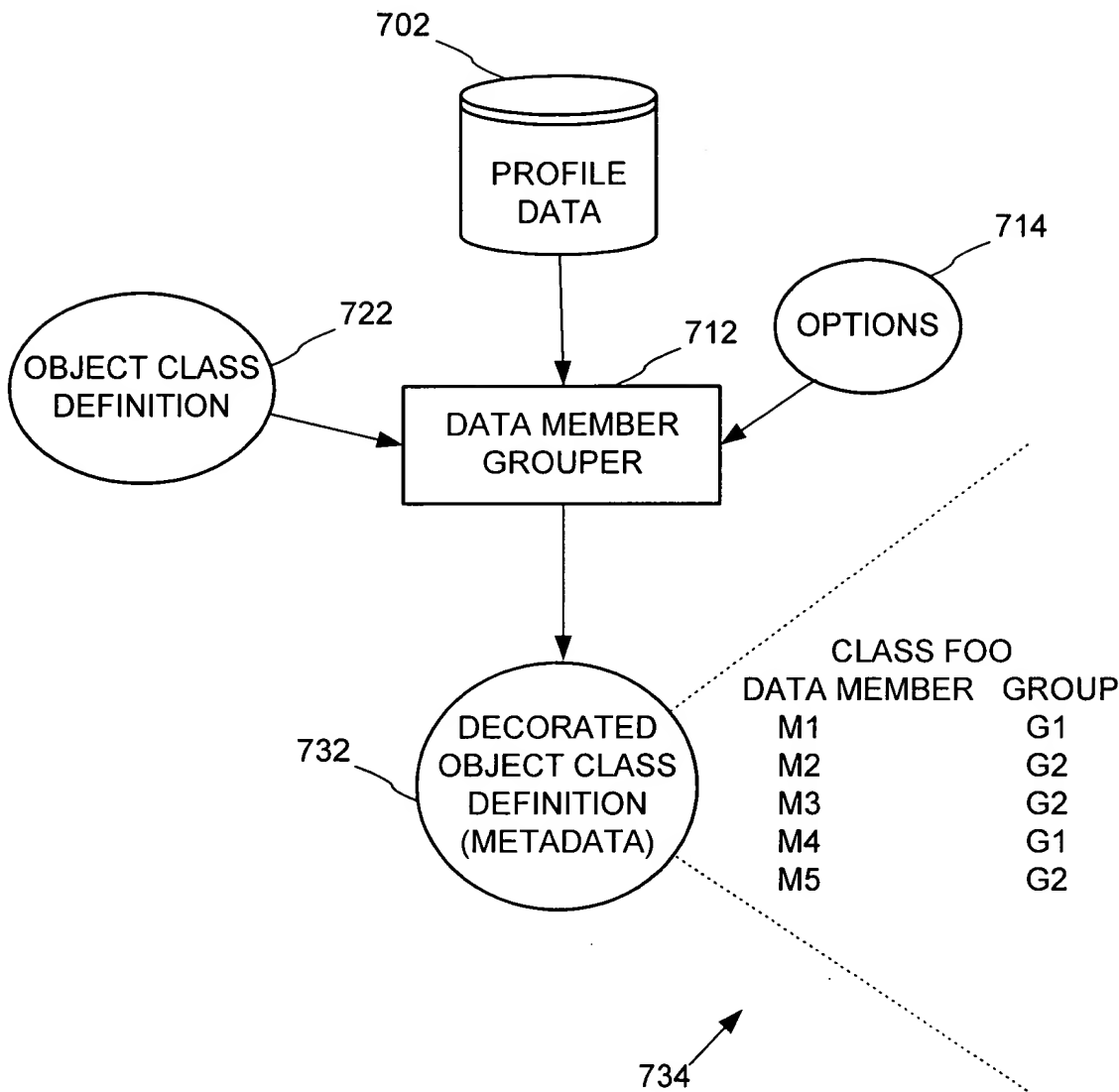


FIG. 8

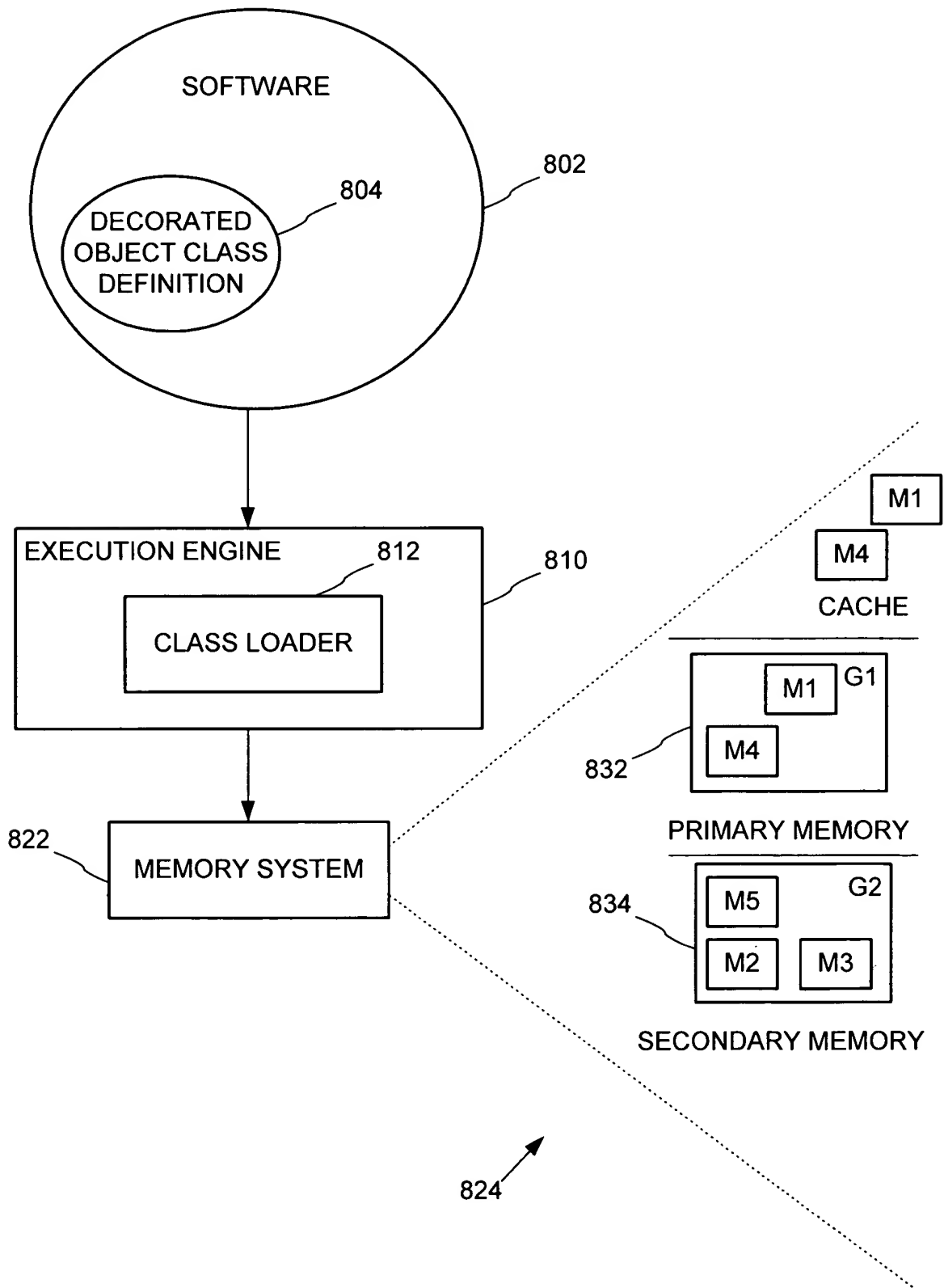
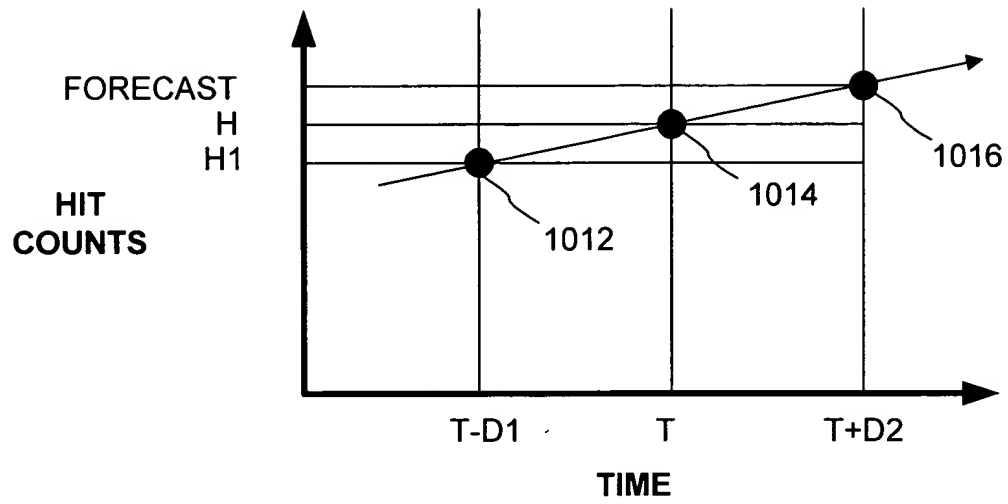


FIG. 9

```
void SaveHitCountsAndPossiblyFlushRecord ()
{
// H = hits so far (overall count)
//H1 = previous value of H
// L = Hits observed during last segment
//      (e.g., method execution)
  if ((H + L) < H) // overflow this time
    FlushRecord () ;
  else
  { H+=L ; //bump overall count by current counts
    L = 0 ;
    // will we overflow next time? (linear forecast)
    if ( ((H - H1) + H) < H)
      FlushRecord () ;
    else
      H1 = H ; // remember current count
  }
}

void FlushRecord ()
{ Write (H, fieldName, className) ;
  H1 = 0 ; // zero out the previous count
  H = L ;
  L = 0 ;
}
```

FIG. 10



1002

FIG. 11

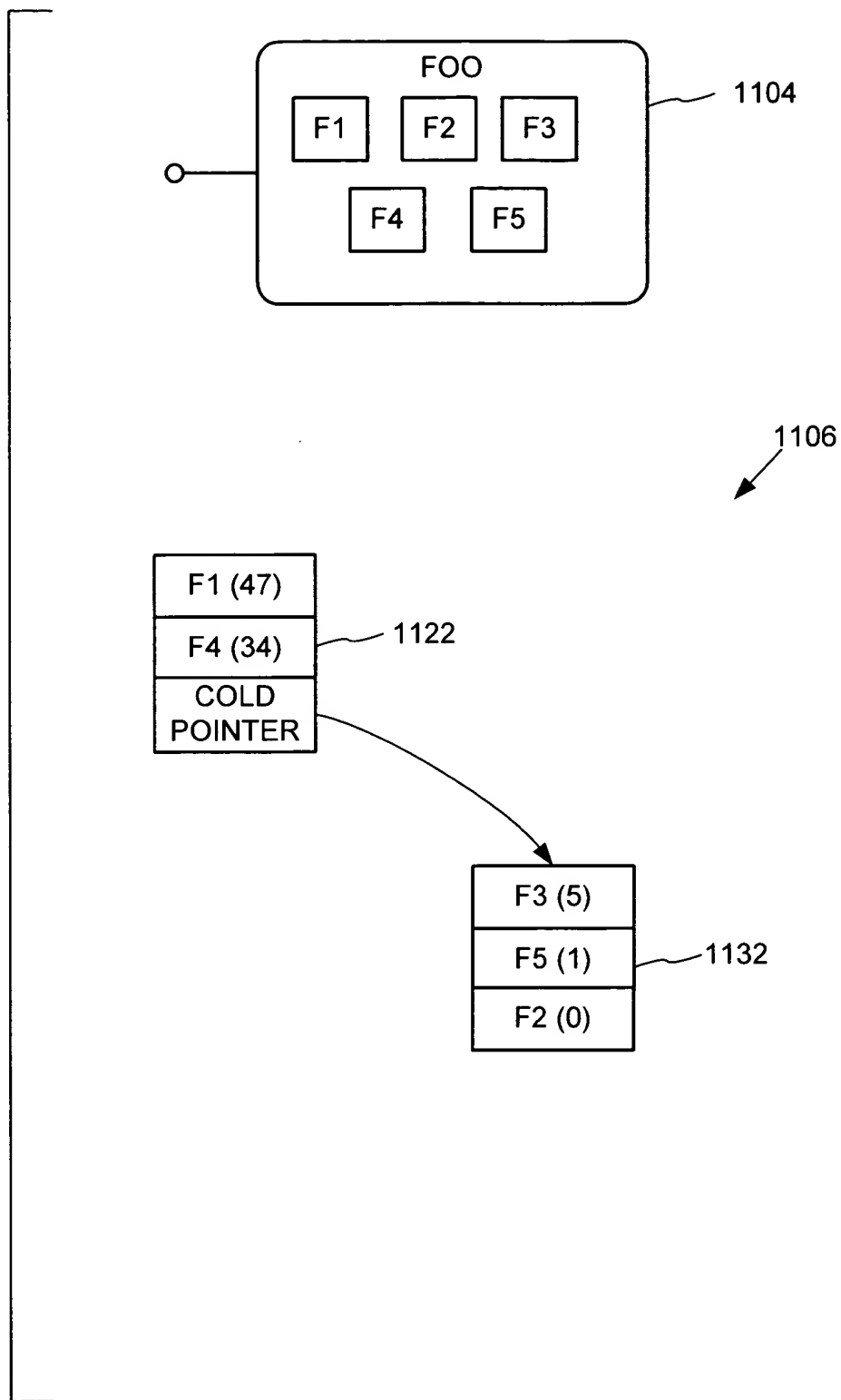


FIG. 12

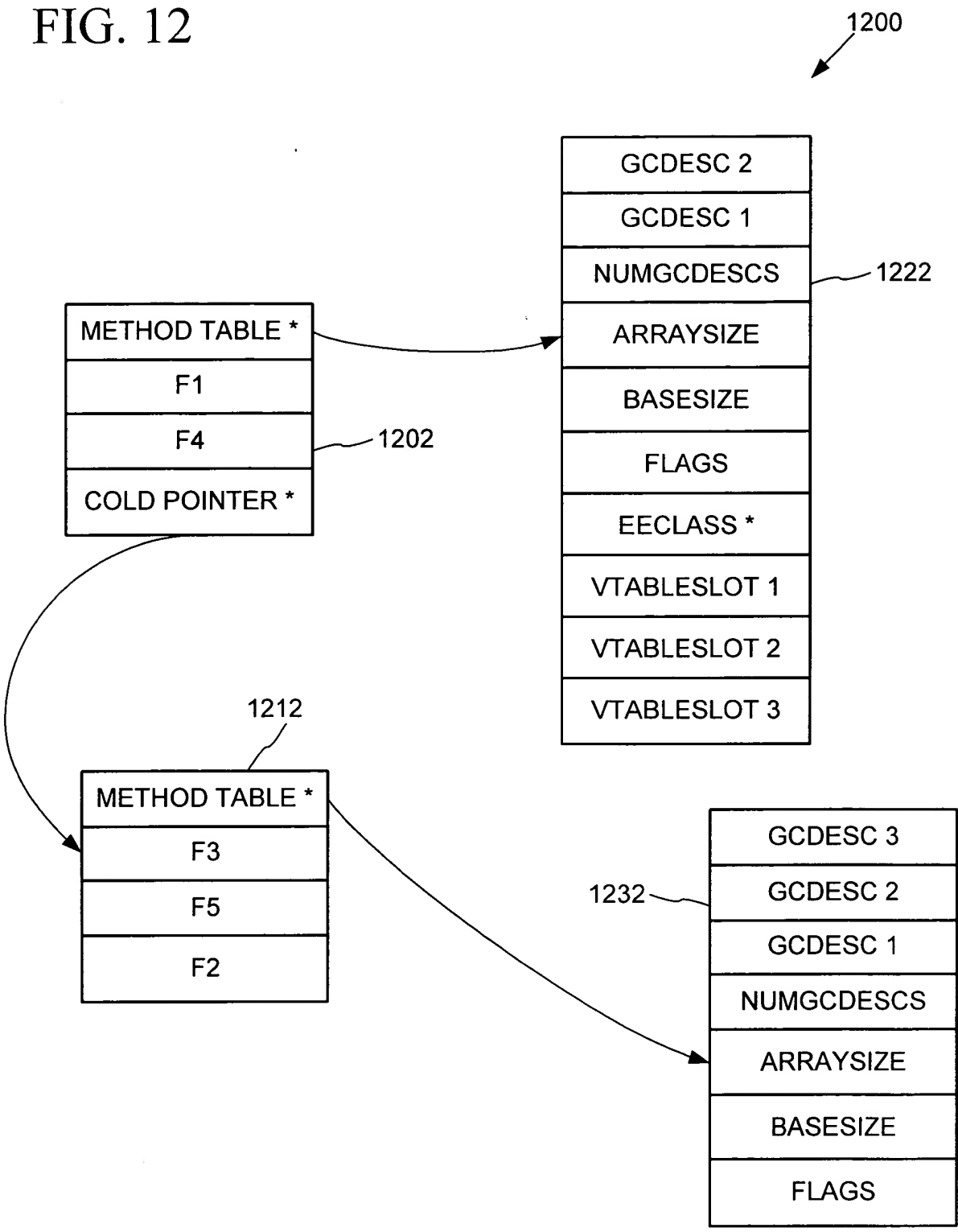


FIG. 13

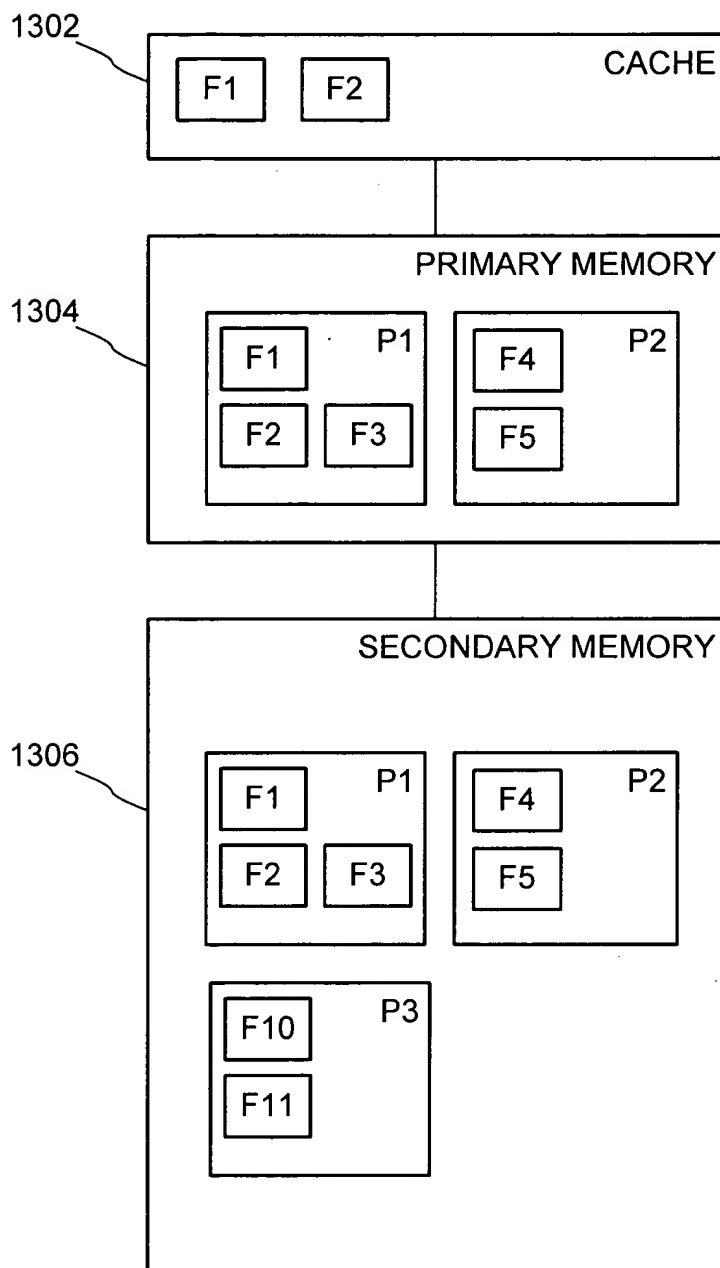


FIG. 14

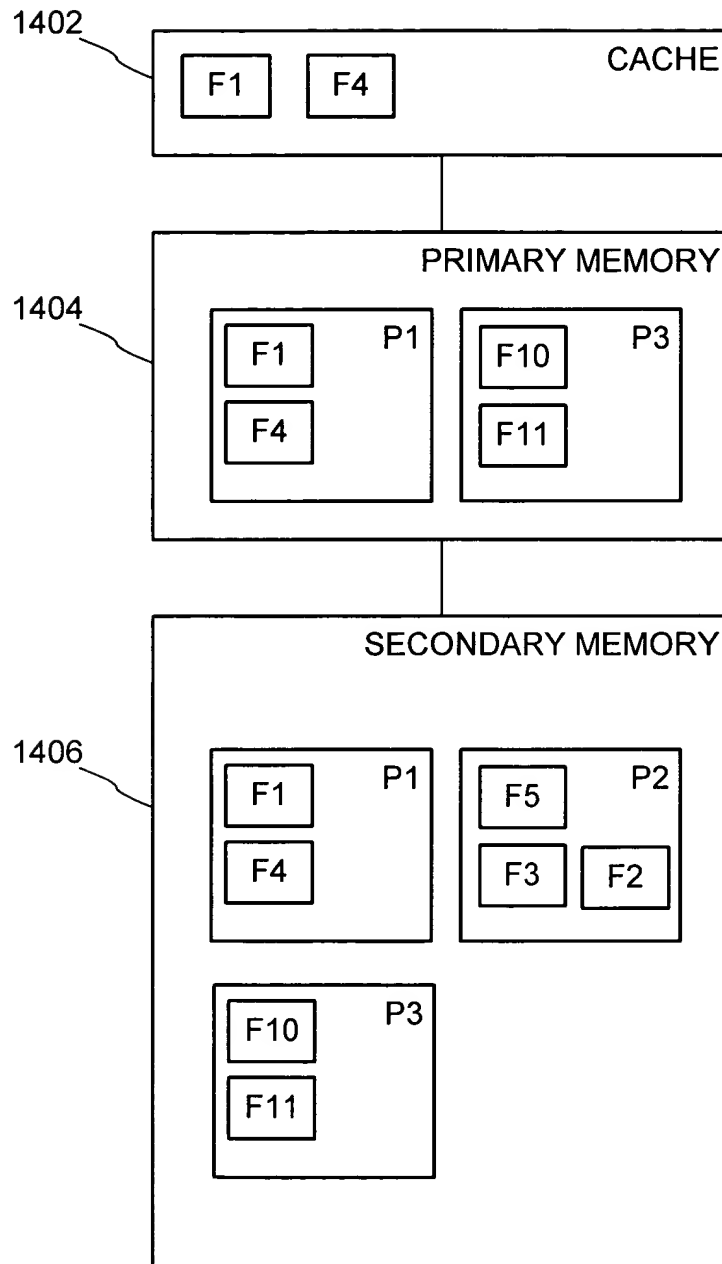


FIG. 17

